



UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII-PATOS-PB
CENTRO DE
CURSO DE GRADUAÇÃO LICENCIATURA EM COMPUTAÇÃO

THYAGO PABLO OTAVIANO VIEIRA

**Contribuições do ensino de Programação em escolas públicas para
a aprendizagem e para o ambiente escolar**

Patos-PB

2015

Contribuições do ensino de Programação em escolas públicas para a aprendizagem e para o ambiente escolar

Trabalho apresentado para conclusão do Curso de Graduação em Licenciatura em Computação da Universidade Estadual da Paraíba.

Orientador (a): Prof. PhD. Rodrigo Alves Costa

PATOS-PB

2015

V658c Vieira, Thyago Pablo Otaviano
Contribuições do ensino de programação em escolas públicas
para a aprendizagem e para o ambiente escolar [manuscrito] /
Thyago Pablo Otaviano Vieira. - 2015.
51 p. : il.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação)
- Universidade Estadual da Paraíba, Centro de Ciências Exatas e
Sociais Aplicadas, 2015.

"Orientação: Prof. Dr. Rodrigo Alves Costa, CCEA".

1. Ensino de programação. 2. Informática educativa. 3.
Aprendizagem significativa. 4. Ensino público. I. Título.

21. ed. CDD 371.334

THYAGO PABLO OTAVIANO VIEIRA

Thyago Pablo Otaviano Vieira

Contribuições do ensino de Programação em escolas públicas para a aprendizagem e para o ambiente escolar

Trabalho de Conclusão de Curso apresentado ao Curso de Licenciatura em Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Licenciado em Computação

Aprovado em 19 de junho de 2015

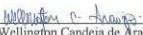
BANCA EXAMINADORA



Rodrigo Alves Costa
(Orientador)



Rosângela de Araújo Medeiros
(Examinador)



Wellington Candeia de Araújo
(Examinador)

SUMÁRIO

1. Considerações Iniciais.....	11
1.2. Objetivos.....	12
1.2.1. Objetivo Geral.....	12
1.2.2. Objetivos Específicos.....	12
1.3. Justificativa.....	13
1.4. Motivação.....	13
1.5. Metodologia.....	14
2. Referencial Teórico.....	15
2.1. Aprendizagem Significativa.....	15
2.2. Informática na Educação.....	20
2.3. Breve Histórico da Programação.....	21
2.4. Ensino de Programação.....	23
2.4.1. Programação e o desenvolvimento de Habilidades.....	25
2.4.2. Ensino de Programação nas Universidades.....	26
2.4.3. Programação nas Escolas.....	31
2.4.3.1. Linguagem Educativa: Logo.....	31
2.4.3.2. Linguagem Educativa Scratch.....	33
2.4.4. O Estudo de Programação Relacionado a Outras Aprendizagens.....	35
2.4.4.1. O Estudo de Programação e o Desenvolvimento de Habilidades Úteis no Processo de Aprendizado.....	36
3. O Ensino de Programação e Relações com a Educação.....	38
3.1. O Ensino de Programação para a Efetivação do Uso dos Laboratórios	38
3.2. A Programação e a Aprendizagem Significativa.....	43
4. Considerações Finais.....	45
Referências.....	48

Lista de Figuras

Figura1. Ranking das Linguagens de Programação.....	30
Figura2. Ambiente do Kturtle.....	33
Figura3. Categorias de funcionalidades do Scratch.....	34
Figura4. Blocos de comandos do Scratch conectados.....	35

Lista de Tabelas

Tabela1.Disciplinas de Ciências da Computação da Universidade Federal de Campina Grande.....	26
Tabela2.Disciplinas Relacionadas a Computação do Curso de Ciências dada Computação da Universidade Federal do Maranhão.....	28
Tabela3.Cargos e graduações exigidas para o concurso para professor do estado da Paraíba em 2012.....	40
Tabela3.Disciplinas obrigatórias do curso de Licenciatura em Ciências da Computação da UFPB.....	42

Contribuições do ensino de Programação em escolas públicas para a aprendizagem e para o ambiente escolar

VIEIRA, Otaviano Pablio Thyago¹

¹ Thyago pablio Otaviano Vieira, graduando em licenciatura em computação pela Universidade Estadual da Paraíba

RESUMO

Este trabalho discute sobre o ensino de programação e busca mostrar as suas contribuições para os alunos e o ambiente escolar. Para comprovar isso, foi realizada uma pesquisa bibliográfica, baseada em artigos de eventos de computação e educação, além de estudos em livros de pesquisadores das áreas relacionadas a esse trabalho, tais como: José Armando Valente, Marc Prensky e David Ausubel. A partir das pesquisas bibliográficas, serão apresentados vários aspectos do ensino de programação, incluindo: a efetivação da utilização dos laboratórios de informática e aprimoramento de capacidades úteis para o aprendizado de outras matérias em sala de aula, a utilização da aprendizagem significativa para o ensino de programação e sua relação com a aprendizagem de outras disciplinas.

PALAVRAS-CHAVE: ensino de programação, informática educativa, aprendizagem significativa, ensino público.

1. Considerações Iniciais

Atualmente o uso do computador está cada vez mais presente na vida das pessoas, e através dele são desempenhadas as mais diversas atividades, sejam elas profissionais, de comunicação, de lazer, entre outras. Os jovens estão entre os maiores usuários, começando a usar o computador desde cedo, muitas vezes desde criança, passando muitas horas fazendo uso de tal ferramenta. Segundo pesquisa do IBOPE (2013), jovens entre 12 à 17 anos passam um tempo médio de 30 horas e 30 minutos usando o computador e eles somam 11,8% de um total de 46,7 milhões de usuários domiciliares do país.

O governo federal tenta estender o uso da computação para as escolas, para que os alunos façam uso de tal ferramenta de forma educativa, como é pretendido pela implantação do Programa Nacional de Tecnologia Educacional (PROINFO), que a partir do comprometimento das escolas fornecerem a estrutura necessária e de fazerem uso do computador também para o ensino, são implantados laboratórios nelas. A grande maioria das escolas públicas, cerca de 81%, sejam elas estaduais como municipais, possuem laboratórios, e segundo pesquisa do Ministério da Educação (2013), 86,7% das escolas públicas brasileiras o possuem de informática. Segundo pesquisa TIC Educação (2011), 76% das instituições públicas de ensino fazem uso dos laboratórios, sendo assim, 24% não usam esses ambientes. Apesar desse programa, nem todas as escolas possuem laboratórios ou as que tem, muitas não fazem uso de tal ambiente.

Em paralelo a isso, nas escolas públicas atualmente é vista uma dificuldade na aprendizagem, tal fato é ilustrado pelo índice de reprovação de alunos do ensino médio que é de 6,2%, de acordo com Censo Escolar (2013), que mostra também que a porcentagem de abandono é de 1,2%, por mais que aparentemente sejam números relativamente baixos, ainda fazem parte da realidade da educação brasileira.

Dessa forma, a educação brasileira ainda apresenta alguns problemas, embora sua incidência seja baixa, porém, elas ainda existem, como foram representadas pelos índices acima apresentados, que foram: a reprovação de série e a evasão escolar. Sendo que, um dos motivos para que isso ocorra pode surgir da própria sala de aula, que pode ser a falta de interesse dos discentes em estudar os conteúdos transmitidos em sala, causando péssimos resultados em avaliações realizadas pelo professor.

Dessa forma, é preciso encontrar maneiras para que o ensino desperte o interesse e conseqüentemente facilite o aprendizado dos alunos. Assim, nesse trabalho é proposto uso do

computador para o ensino de disciplinas escolares pode ser uma opção para contribuir no aprendizado dos discentes, pois, a maioria das escolas possuem laboratórios e outra coisa é que boa parte dos discentes já fazem uso dessa ferramenta em casa, o que mostra o interesse por tal, o que poderia ser transferido para o aprendizado de matérias que são vistas em sala de aula. Desse modo, também é defendido nessa produção que, isso pode contribuir para o aprendizado com o computador. Assim, os alunos reforçariam os conteúdos vistos em sala e aprenderiam novas funcionalidades do computador, aliando a computação com a parte educativa.

A questão é responder como exatamente o computador se aliaria ao ensino de disciplinas escolares e como os alunos reforçariam aquilo que já estudaram em sala e ao mesmo tempo aprenderiam conceitos novos acerca da computação. O ensino de programação, pode ser uma proposta para auxiliar no ensino de matérias escolares, como já vem sendo colocado em prática em alguns ambientes, tanto em escolas públicas como em privadas, mas não de forma efetiva. Algumas escolas em São Paulo, oferecem o ensino dessa disciplina, como a escola Bakhita, que oferta tanto a programação, como a robótica também, porém, é uma matéria extra-curricular, destinada a alunos da faixa-etária de 8 a 13 anos de idade.

A Escola Bakhita, onde Artur estuda, reúne semanalmente alunos de 8 a 13 anos para aprender programação e robótica em aulas práticas. Oferecidas como atividade extracurricular, as aulas seguem uma tendência mundial que ganha força no Brasil, o incentivo ao ensino da linguagem da computação em escolas.

(AGUILHAR, Lígia. Escolas e Clubes Ensinam Crianças a Programar. Em: <http://blogs.estadao.com.br/link/escolas-e-clubes-ensinam-criancas-a-programar/>)

A proposta desse trabalho é expor as contribuições que a efetivação desse tipo de disciplina pode ter na aprendizagem dos alunos, como também no ambiente escolar. Para isso serão analisados fatores a seguir: como a aprendizagem significativa pode colaborar para que o aluno aprenda de fato programação e outras disciplinas; como o ato de programar pode despertar e aprimorar habilidades importantes, que podem ser importantes para o aprendizado; como o ensino de programação pode contribuir para a efetivação do uso dos laboratórios de informática nas escolas públicas e como tudo isso pode contribuir para que o ensino de programação seja efetivado nas escolas públicas.

1.2.Objetivos

1.2.1.Objetivo Geral

Este trabalho tem como objetivo geral mostrar como o ensino de programação em escolas públicas pode contribuir de diversas formas, tanto para o aprendizado do aluno, como para o ambiente escolar.

1.2.2.Objetivos específicos

Os objetivos específicos deste trabalho são:

- Mostrar a aprendizagem significativa pode contribuir no ensino de programação;
- Mostrar que o ensino de programação, pode contribuir para a efetivação do uso dos laboratórios de informática nas escolas públicas;
- Mostrar se o ensino de programação pode aprimorar capacidades que podem ser usadas na aprendizagem de determinada disciplina escolar.

1.3.Justificativa

Devido a um grande desinteresse dos alunos em fase escolar por algumas disciplinas, resultando na não compreensão dos conteúdos em sala de aula, o uso da informática é visto como uma alternativa para auxiliar na compreensão dos conteúdos pelos alunos. A solução para esse problema pode contribuir com outros problemas, como: a falta de efetivação da utilização dos laboratórios de informática e de programação nas escolas públicas. Sendo assim, com a importância e as contribuições do ensino de programação sendo expostas, a intenção é que o uso dos laboratórios, como o ensino de programação sejam efetivados nas escolas públicas.

1.4.Motivação

A motivação para este trabalho é o do atual desinteresse dos alunos em fase escolar em relação as disciplinas que eles estudam, que ficou evidenciado nos índices de evasão e reprovação escolar, apresentado no item 1. Sendo assim, é percebido que uma alternativa para que motive e desperte o interesse dos alunos em relação ao aprendizado em disciplinas escolares é a utilização dos laboratórios que muitos ambientes de ensino público dispõem para o ensino de matérias vistas em sala de aula através da instrução da programação. Assim, é

defendido nesse trabalho que os discentes poderão aperfeiçoar seus conhecimentos sobre os assuntos de outros componentes curriculares, como também aprender conceitos de programação. Além disso, também é defendido que, a partir do estudo dessa matéria, os alunos podem desenvolver ou aprimorar habilidade úteis no processo de aprendizagem.

1.5. Metodologia

O trabalho foi feito a partir de uma pesquisa bibliográfica. A pesquisa bibliográfica foi enfatizada em artigos e livros, que fossem relacionados as várias áreas fazem parte do tema, como: educação, psicologia e tecnologia. A coleta de informações foi feita no período entre Abril e Setembro de 2014, e foram buscados trabalhos com os seguintes temas: aprendizagem significativa, história da programação, ensino de programação em escolas, contribuições do ensino de programação nas escolas, desenvolvimento de habilidades úteis para a aprendizagem em sala de aula. Foram procuradas publicações que estivessem ligadas direta ou indiretamente com a proposta deste trabalho.

A pesquisa bibliográfica, foi necessária para a realização desse trabalho, para ter um contato maior e mais aprofundado com todo o conteúdo e de diversas áreas que abrangem a monografia. Para que assim, o pesquisador com base nos resultados dessa pesquisa, terá base para desenvolver seu trabalho.

Para Marconi e Lakatos (1992, apud PERSKE, 2004, p. 11), a pesquisa bibliográfica é o levantamento de toda a bibliografia já publicada, em forma de livros, revistas, publicações avulsas e imprensa escrita. A sua finalidade é fazer com que o pesquisador entre em contato direto com todo o material escrito sobre um determinado assunto, auxiliando o cientista na análise de suas pesquisas ou na manipulação de suas informações.

Os artigos foram pesquisados no período determinado anteriormente e buscados nos anais de eventos relacionados a computação, como: Encontro Nacional de Pesquisa em Educação em Ciências(2009-2013), Congresso da Sociedade Brasileira de Computação(2009-2013) e do Simpósio Brasileiro de Informática na Educação(2009-2013).

Neste trabalho foram utilizadas algumas práticas realizadas por outros autores, para justificar, que o ensino de programação pode ser de forma significativa e que o ensino dessa disciplina pode colaborar o aprendizado em outras disciplinas. A partir dos resultados das pesquisas, buscou-se mostrar as mais diversas contribuições do ensino de programação relacionadas a aprendizagem dos alunos, como também para o ambiente escolar. Além disso,

a própria pesquisa bibliográfica tem sua importância para que, a partir do que foi pesquisado, seja possível construir expor uma base teórica e em seguida construir argumentos sólidos, para que possam comprovar o que é defendido nesse trabalho.

2. Referencial Teórico

Para realização desse trabalho, foram realizadas pesquisas focadas em vários assuntos relacionados ao tema, como: o ensino de programação em escolas, nas universidades, sua contribuição para desenvolver e aprimorar habilidades úteis ao aprendizado escolar e aprendizagem significativa. A pesquisa foi feita em artigos e trabalhos, que abordem assuntos relacionados ao tema do trabalho, além de coleta de artigos de eventos de computação.

A partir de todo o material que foi pesquisado, procurou-se conhecer parte do que já tinha sido estudado sobre aprendizagem significativa, sobre as capacidades despertadas através do exercício da programação, sobre efetivação dos laboratórios em sala de aula. Buscou-se também contribuir para a literatura com informações adicionais, com o objetivo de alcançar os objetivos gerais e específicos deste trabalho.

2.1. Aprendizagem significativa

Moreira e Mazini (1982), ao problematizar e explicar a teoria da aprendizagem significativa de David Ausubel, abordam conceitos relacionados a ela, como: cognição, cognitivismo e subsunçores. Assim, explicando como vários elementos compõem esse tipo de aprendizagem. Os autores explicam como esse tipo de aprendizagem acontece, quais os fatores que a tornam favoráveis e quais elementos que estão envolvidos nesse processo.

Criada por David Ausubel, a teoria da aprendizagem significativa é uma proposta de método de ensino que, basicamente, sua ideia era de, durante o processo de aprendizagem, os alunos simplesmente recebessem conteúdos que pudessem ser associados a algo que eles já saibam, seja do seu cotidiano ou mesmo do que já foi aprendido anteriormente. Em seu processo, ela faz uso, portanto, de alguns elementos, como: consciência, intencionalidade e cognição.

Consciência é responsável por dar sentido e significado às coisas com as quais o indivíduo tem contato, para que, assim, torne-se possível entender melhor o ambiente em que se vive, atribuindo inclusive significados a outros objetos.

É a consciência que atribui significados aos objetos que rodeiam o indivíduo. Sendo assim, o sujeito através da consciência, passa a compreender melhor o mundo, podendo aprimorá-la, a partir de sua própria compreensão, dando significado a mais objetos e a partir de alguns significados acabam surgindo outros (MOREIRA; MASINI, 1982).

Dessa forma, os seres humanos passam a dar significados às coisas ao seu redor através da consciência, e dessa forma passam a compreender melhor o meio onde vivem. Com o decorrer do tempo e com uma melhor compreensão do mundo, é possível passar a criar novos significados, que possam ser associados a novas coisas.

O processo ao qual a consciência acontece, ou seja, quando os significados das coisas que o indivíduo tem acesso começam a surgir, sendo que os significados podem ser o ponto de partida para outros significados (MOREIRA; MASINI, 1982).

Assim, o indivíduo não apenas obtêm os conhecimentos, mas passam a entender que eles podem ter outros significados, além daqueles ensinados, como também outras finalidades.

A intencionalidade possui uma forte ligação com a consciência, é a estrutura do sentido que nos possibilita, sujeitos que somos de vermos e compreendermos o mundo. Dessa forma, a partir dos significados que são atribuídos as experiências ao longo de sua vida, o ser humano vai aprendendo quais medidas tomar, como agir diante de tal situação, dando sentido as suas ações. Sendo assim, tudo pelo qual o indivíduo passa acaba o mostrando quais ações deve tomar. (MASINI; MOREIRA, 1982)

A intencionalidade, como a consciência, também é responsável por atribuir significados às coisas que estão ao redor do ser humano, o que acaba incluindo as ações realizadas por eles e contribuindo na forma como e porque as realizam.

Relacionado à origem do mundo de significados, ou seja, à definição de sentidos daquilo que o indivíduo tem acesso, existe a cognição, que é o processo através do qual o sujeito, de acordo sua relação com o mundo ou com o meio qual ele tem acesso, começa a atribuir significados e, por conseguinte, acaba dando origem a outros.

A medida que o ser humano se situa no mundo, estabelece relações de significação, isto é, atribui significados a realidade que se encontra, esses significados não são entidades estáticas, mas pontos de partida para a atribuição de outros significados (MOREIRA; MASINI, 1982, p.3).

Todos esses significados que são adquiridos através do contato do ser humano com o mundo precisam de um lugar para ser armazenados, da mesma forma que um computador

precisa guardar o que for executado pelos usuários. O responsável pelo armazenamento de tudo que é aprendido pelo sujeito, é a estrutura cognitiva.

É a estrutura cognitiva, entendida como conteúdo total de conhecimentos de um certo indivíduo e sua organização; ou conteúdo e organização de suas ideias em uma área particular de conhecimentos (AUSUBEL, 1968, apud MOREIRA; MASINI, 1982).

Aprendizagem, segundo Ausubel (1968, apud MOREIRA; MASINI, 1982), significa como estão organizados e integrados na estrutura cognitiva. Isso quer dizer que aprendizagem remete a forma como conceitos estão dispostos e como eles se interligam estrutura cognitiva. A aprendizagem significativa é, portanto, um processo que interliga, na estrutura cognitiva, novas informações com antigas, que tem algum tipo de relação entre elas. Essas informações, que já existem na estrutura cognitiva do ser humano e que têm relação com as novas, são chamadas de subsunçores, que, por sua vez, têm a função de se relacionar com as novas informações, as âncoras, retendo-as (ou “ancorando-as”) na estrutura cognitiva do indivíduo.

Os seres humanos, principalmente aqueles que estão sendo submetidos a algum tipo de aprendizagem, recebem informações de forma constante, tendo como objetivo retê-las em suas estruturas cognitivas. Essas informações, para que realmente sejam retidas, precisam se relacionar com conteúdos que já estejam em tal região, os subsunçores, e ter alguma ligação com eles. A aprendizagem significativa seria, portanto, uma continuação da aprendizagem, ampliando o significado daquilo que está sendo aprendido a partir da relação com o que já estava na estrutura cognitiva.

A aprendizagem mecânica (*role learning*) como sendo a aprendizagem de novas informações com pouca ou nenhuma associação com conceitos já existentes na estrutura cognitiva. (AUSUBEL, 1968, apud MOREIRA; MASINI, 1982)

Na disciplina de programação, para se ensinar sobre, por exemplo, declarar variáveis, é necessário que os alunos tenham em sua estrutura cognitiva alguns conceitos que colaborarem para o novo entendimento. Um desses conhecimentos prévios seriam os identificadores e suas funções, para que assim possa ser possível associá-los às variáveis. Nesse caso, os conceitos acerca de identificadores seriam os subsunçores necessários para o aprendizado de declaração de variáveis.

Silva e Tavares (2012) buscam, a partir da utilização do Scratch, trabalhar os conhecimentos que os alunos já tenham, para assim reforçá-los. Trabalharam as disciplinas de

Matemática e Literatura, estudando matérias como plano cartesiano e literatura de cordel. O objetivo dessa prática, foi o ensino animações no Scratch, usando de base a literatura de cordel, assuntos esses que estavam sendo vistos em sala de aula. Os conhecimentos sobre cordel que os alunos iriam precisar para desenvolver suas atividades foram passados em sala de aula e através de pesquisa. Em seguida, foram realizadas leituras de poemas, como também foram estudados as estruturas e características do cordel. Após essas etapas, os discentes criaram seus próprios textos (SILVA; TAVARES, 2012).

A próxima tarefa foi desenvolver um jogo sobre cordel, baseado nos textos dos alunos, usando de uma das características do cordel, que o é uso de imagens em preto e branco.

A ideia era fazer uma pista em fundo monocromático utilizando a programação disponível do Scratch. Em seguida, teriam que buscar (importar) a figura de um meio de transporte e acoplá-lo ao programa. Os comandos eram dados para que colocassem o transporte em um ponto inicial. Em seguida, era pedido para que programassem os comandos de setas para que o carrinho se movimente para cima; para baixo; para frente e para trás; segundo as setas do teclado do computador. Para que houvesse avanço, outro comando foi dado para que os transportes movimentassem apenas dentro da pista e se se encostassem às laterais ou saíssem da pista deveriam programar e novamente voltar para ponto inicial (SILVA; TAVARES, 2012)

Nessa prática de ensino, o trabalho com a literatura de cordel serve para auxiliar o aprendizado do Scratch, ocorrendo dessa forma interdisciplinaridade entre a literatura e a programação, pois para aprender conceitos referentes a programação, que serão utilizados para fazer a animação, no ambiente de desenvolvimento, que foi pedido pelo professor para que os alunos fizessem, é necessário que os alunos entendam e aprendam sobre literatura de cordel.

Para que a aprendizagem significativa aconteça, é fundamental que os alunos já tenham conhecimento prévio acerca do assunto a ser explicado em sala de aula ou acerca de assuntos similares, que contribuíssem para que o discente assimilasse o que fosse aprendido. Segundo Ausubel (1968 apud MOREIRA; MASINI, 1982), aprendizagem significativa é um processo pelo qual uma nova informação se relaciona com uma antiga presente na estrutura cognitiva do indivíduo, tornando mais simples a tarefa de assimilar os novos conhecimentos.

No caso de conceitos aprendidos anteriormente que contribuem para a criação de uma noção sobre os novos conteúdos podem contribuir para a criação de subsunçores. Tais conceitos são denominados de organizadores prévios (MOREIRA; MASINI, 1982). Segundo Moreira e Masini (1982), organizadores prévios são materiais que contêm conceitos básicos

sobre o que será estudado. No caso da prática de Silva e Tavares (2012), os conhecimentos sobre literatura de cordel poderiam ser considerados como organizadores prévios, pois além de serem materiais expostos antes do ensino do conteúdo, contêm o que será explicado mais adiante de forma generalizada e utilizando termos familiares para o aluno.

Os organizadores são mais eficientes quando apresentados no início das tarefas de aprendizagem, do que quando introduzidos simultaneamente com o material aprendido, pois dessa forma suas propriedades integrativas ficam salientadas. Para serem úteis, porém, precisam ser formulados em termos familiares aos alunos, para que possam ser aprendidos e devem contar com boa organização do material de aprendizagem para terem valor de ordem pedagógica. (MOREIRA; MASINI, 1982)

A produção de organizadores prévios, para uma conseqüente criação de subsunçores, caracterizaria uma aprendizagem significativa. Para que ela de fato ocorra, Moreira e Masini (1982), defendem que devem ser atendidas duas condições:

- O que será aprendido possa ser relacionado a estrutura cognitiva de forma não arbitrária;
- O aluno precisa estar disposto a relacionar o que está aprendendo e o que já foi aprendido, de forma não arbitrária.

É importante destacar que o segundo ponto refere-se ao interesse do aluno em aprender, pois, obviamente, não importa se a forma de aprendizagem é significativa ou se existe todo um mecanismo que favoreça o ensino de forma fácil o conteúdo trabalhado se o aluno não quiser realmente aprender. De acordo com Prensky (2008), o papel da tecnologia nas salas de aula é o de apoiar os alunos no aprendizado. Os alunos, com a utilização das tecnologias, obtêm conhecimento de forma rápida e prática, sendo também responsáveis pelo seu próprio ensino. Os professores que anteriormente exclusivamente transmitiam os conteúdos, passam agora a atuar como auxiliares dos alunos. Os alunos, que antes apenas aprendiam o que professores lhes passavam, hoje têm a oportunidade de construir seu próprio conhecimento.

2.2. Informática na educação

Fugimoto (2009) aborda o tema da informática na educação, expondo no início que o uso do computador está em ascensão, chegando nos mais variados ambientes, como a escola.

Não basta a escola adquirir recursos tecnológicos, é necessário ter professores capazes de atuar, de refletir e de criar ambientes de aprendizagem na busca de contribuir para o processo de mudança do sistema de ensino. (FUGIMOTO, 2009)

Além de o professor utilizar o computador, a sua metodologia tem que atender as necessidades dos alunos que, por se encontrarem na atual sociedade da informação, são mais críticos e criativos. Essa criticidade vai muito além do ambiente tradicional de sala de aula e para uma aprendizagem verdadeiramente significativa, a utilização do computador provavelmente necessitará incorporar usos do cotidiano dos escolares, para que o simulacro computacional faça sentido organizacionalmente. Assim, não basta utilizar o computador e as melhores ferramentas disponíveis em sala de aula, o aspecto criativo tem fundamental importância no aspecto da promoção de aprendizagem verdadeiramente significativa.

Quanto ao professor, para utilizar o computador como ferramenta educacional, necessita de uma formação que o capacite para o desempenho de atividades que contribuam para a construção do conhecimento do aluno. (FUGIMOTO, 2009)

Não basta, portanto, que a escola tenha um professor que utilize o computador em suas aulas, ele precisa ter qualificação para tal, para a utilização das ferramentas e principalmente, sobre como utilizá-la de forma educativa. Por sua vez, Prensky (2008) elucida o papel da tecnologia nas salas de aula, que é o de apoiar os alunos no aprendizado de certa forma os ensinando. Os alunos, com a utilização das tecnologias, obtêm conhecimento de forma rápida e prática, sendo também responsáveis pelo seu próprio ensino.

A tecnologia de hoje, no entanto, oferece aos alunos todos os tipos de ferramentas novas e altamente eficazes que podem usar para aprender por conta própria - a partir da Internet com quase toda a informação, de pesquisa e ferramentas de pesquisa para separar o que é verdadeiro e relevante, para ferramentas de análise para ajudar a dar sentido a ela, para ferramentas de criação para apresentar as conclusões de uma em uma variedade de meios de comunicação social, às ferramentas sociais para a rede e colaborar com as pessoas ao redor do mundo (PRENSKY, 2008).

Nesse novo desenho do papel de professores e alunos, os discentes obtêm conhecimentos de forma mais rápida. No entanto, os mesmos precisam saber separar

informações verdadeiras, daquelas que são falsas, para evitar equívocos quanto às informações obtidas. Ou seja, invariavelmente, o professor deve ser consultado, o que mostra que, mesmo com o uso do computador para obter conhecimento, a importância do professor ainda existe.

2.3. Breve histórico da programação

A prática de escrever instruções, ou seja, ações para que uma máquina a interprete e assim, alcançar os objetivos pretendidos pelo usuário; pode ser datado do século XIX, a partir da invenção de Jackard. Em séculos passados eram construídas máquinas mecânicas, que tinham a função de resolver cálculos matemáticos. Com o passar dos tempos seus inventores, viram a necessidade de automatizar suas tarefas, tornando a execução determinadas ações, mais rápidas e precisas.

Em 1801, Joseph Marie Jackard (1752-1834), desenvolveu um tear mecânico, que tinha como objetivo tornar o processo de tecelagem mais rápido e automatizado, para isso, o aparelho era controlado por cartões perfurados. Os furos dos cartões possuíam os comandos necessários para que os fios, usados para fazer a peça desejada, fossem movidos ou não. Assim, tomavam a forma desejada.

Os buracos nos cartões que controlam mecanicamente quais os conjuntos de fios que se movem ou não, de modo a criar o modelo desejado. Quando um dos cartões se posiciona em oposição e é pressionado contra o conjunto de varas, esses fios das varas opostas a um buraco no cartão permanecem no lugar, enquanto os outros fios são movidos (SOARES, 2010).

Ao longo dos anos, várias máquinas que existiram, principalmente máquinas que tinham funções de realizar cálculos, utilizavam cartões perfurados. Um exemplo do uso desse tipo de programação primitiva seria a máquina de diferenças, de Charles Babbage (1792-1891), que ele tentou construir em 1822. Ele tinha como principal ideia, que as máquinas poderiam manipular informações, caso fosse possível convertê-las em números. A partir dessa ideia, o projeto desse equipamento foi se desenvolvendo, que teria como função resolver equações polinomiais por meio do cálculo de sucessivas diferenças entre conjuntos de números. A entrada de dados para que fosse possível a execução da função do invento, seria feita por cartões perfurados.

A entrada de dados para a máquina seria feita através de três tipos de cartões: “cartões de números”, com os números das constantes de um problema; “cartões diretivos”, para o controle dos movimentos dos números na máquina; e “cartões de operação” para dirigir a execução das operações, tais como adições, subtrações, etc (FILHO, 2007).

Nessa máquina, existiam cartões perfurados, que possuíam determinadas funções importantes para que ela executasse ações e também que armazenavam dados que seriam utilizados na resolução de problemas.

Ada Lovelace (1815-1852), conhecida como a primeira programadora, chegou a escrever programas para essa máquina, que não chegou a ser finalizada. Dentro desses escritos haviam muitas estruturas de programação que são usadas até hoje, como: loops, subrotinas e saltos.

Quando, na primeira metade do século XX, surgiram os primeiros computadores que utilizavam de energia elétrica, a exemplo do ENIAC e do UNIVAC. A forma de programar também mudou para representar comandos e ações, e passaram a ser utilizados caracteres, como letras, números e símbolos. Para representarem os comandos dos programas, eles eram passados para fitas magnéticas e o computador interpretava o programa.

Esse tipo de linguagem que surgiu por volta dos anos 40, ficou conhecida por linguagem de máquina ou de baixo nível. Um exemplo típico é a linguagem Assembly, que ficou conhecida dessa forma pois o programador executava o programa manualmente, a partir da manipulação de fios e cabos, que eram colocados em um painel. Assim, o código era executado pela própria máquina.

Mesmo quando programava com esses códigos de operação mnemônicos (também chamados de linguagem de montagem) o programador ainda estava trabalhando em termos dos conjuntos de instruções da máquina, isto é, os algoritmos eram expressos em termos de instrução muito primitivas (detalhes sobre registradores, endereços e saltos, etc). Daí a denominação linguagens de baixo nível. (FILHO, 2007).

Pela grande dificuldade de entender e compreender os códigos de baixo nível a comunidade computacional entendeu a necessidade de criar uma linguagem que fosse mais fácil de compreender e escrever. Surgiram linguagens cuja codificação se aproximava mais da linguagem humana, as chamadas de linguagens de alto nível. A linguagem C é um exemplo desse tipo de paradigma - suas palavras reservadas são vocábulos próximos da língua inglesa.

Palavras reservadas são pré-definidas na biblioteca de uma linguagem de programação, e cada uma tem um papel específico para poder desenvolver um programa. Segundo Sebesta (2010), palavra reservada, é um vocábulo especial dentro de uma

determinada linguagem de programação, que não pode ser usada como variável. As linguagens de alto nível, por serem mais próximas da linguagem humana, precisam de um programa que receba o código e o torne compreensível para a máquina. Esse programa ficou conhecido como compilador, que passou a ser usado desde então.

o compilador recebe como entrada um arquivo com uma especificação em linguagem de alto nível, o código-fonte e gera como saída um outro arquivo com o programa equivalente em linguagem simbólica do processador de plataforma de destino. Nesse caso, um outro programa, o montador (o assembler), é utilizado para traduzir o código em linguagem simbólica para linguagem de máquina. (RICARTE, 2008)

O compilador é, portanto, um programa que existe no ambiente de desenvolvimento usado pelo programador, que tem como função traduzir o código fonte em linguagem de máquina, para que assim, o computador possa executar as tarefas que software desempenhar.

Foi visto como a programação evoluiu ao longo dos anos, desde quando eram usadas em máquinas mecânica, até o seu uso em máquinas eletrônicas. E nessa evolução, ela passou de códigos feitos em cartões perfurados, a linguagens que cada vez mais se aproximavam com a linguagem natural do ser humano.

2.4. Ensino de Programação

O Processo de Ensino-Aprendizagem de Fundamentos de Programação: Uma Visão Crítica da Pesquisa no Brasil

O artigo expõe as dificuldades que os alunos podem encontrar na disciplina, usando de referência os anais da SBIE E WIE entre os anos de 2009 a 2013. A partir disso são analisados alguns fatores responsáveis por essas dificuldades, como: uso de metodologia tradicional, falta de habilidades prévias necessárias para a resolução de problemas em programação e a utilização ou não de uma linguagem de programação. O uso de metodologia tradicional, por ser um método no qual o professor expõe o seu conteúdo sem muita participação do aluno, acaba desmotivando o aluno.

O modo tradicional não consegue facilmente motivar os alunos a se interessar pela disciplina, pois não é claro para os alunos a importância de certos conteúdos pela sua formação”. (BORGES, 2000, apud PEREIRA, RAPKIEWICZ).

O importante nesse caso não seria simplesmente a exposição do assunto, mas seria necessário que houvessem explicações sobre para que serviria tal assunto na futura vida

profissional do discente. Assim, para facilitar o entendimento dos conteúdos, seria interessante que os docentes façam ligações de um programa ou parte dele, que utilize dos mesmos recursos que estão sendo ensinados.

De acordo com Lemos, Barros e Lopes (2003 apud PEREIRA,RAPKIEWICZ), “há uma forte indicação de que o conhecimento anterior sobre resolução de problemas dentro do contexto anterior à computação é um pré-requisito importante no aprendizado de linguagens de programação”. A utilização ou não de uma linguagem de programação é discutido, como forma para contribuir no ensino de programação. Esse assunto acaba dividindo autores entre o ensino de: algoritmos, linguagens orientadas a objetos e linguagens imperativas.

Andrade et al (2004) têm por objetivo mostrar como alunos do 9º ano do ensino fundamental, podem aprender, ou melhor, despertar o interesse por matemática trabalhando através do ambiente de programação educativa, Scratch. Os alunos aprenderam conceitos relacionados a programação exercitando matemática. Os autores estimularam que os discentes produzissem jogos utilizando a o Scratch e, durante o desenvolvimento dos projetos, que relacionavam o uso da programação com a matemática; os alunos demonstraram interesse e domínio da ferramenta que estava sendo usada.

Quando perguntado sobre se as atividades de produção de jogos com o Scratch estavam ajudando a gostar de matemática outro aluno respondeu: “Acho que sim, esse programa usa muito números e também tenho que saber a resposta certa para quando eu fizer um jogo de pergunta o gatinho dá os parabéns”. (Aluno A) (ANDRADE et al, 2004)

Isso mostra que, durante as atividades desempenhadas com a utilização do scratch, não bastava apenas que os alunos utilizassem os comandos da ferramenta que fossem úteis na realização dos projetos, mas também que eles usassem de conhecimentos matemáticos. E isso contribui para que os alunos passassem a gostar de matemática, pois eles estavam vendo a utilidade dessa disciplina para a produção do jogo que eles fossem fazer.

O Scratch, por ser uma linguagem de programação educativa, faz uso de outras disciplinas para ensinar os seus conceitos e principalmente, para que os alunos as utilizem como base para seus jogos. Nesse caso, a matemática pode ser usada, a partir conteúdos recém aprendidos pelos discentes, para desenvolverem seu games, por exemplo, os educandos podem fazer jogos sobre as 4 operações ou numeros primos, entre outros assuntos.

Assim, a partir do interesse por programar no Scratch, conseqüentemente pode refletir no interesse uso da matemática, pois, diferente da sala de aula onde eles apenas resolvem

fórmulas, no jogo que forem fazer, eles devem dar um motivo para o qual o jogador usará de conhecimentos matemáticos.

2.4.1 Programação e o desenvolvimento de habilidades

Gomes et al (2008) falam sobre dificuldades encontradas pelos alunos em aprender programação, enfatizando as habilidades necessárias para a aprendizagem dessa disciplina, principalmente relacionadas a resolução de problemas, como raciocínio e lógica. Os autores defendem que, ao estudar programação, o aluno, diante de um problema a ser resolvido, deve refletir e assim, compreê-lo, para saber como solucioná-lo. Além disso, é necessária muita prática para que o discente saiba como agir diante dos mais variados tipos de problemas.

. É possível relacionar a resolução de problemas de programação a várias habilidades, como: persistência, para que os alunos não desistam de resolver determinada questão; compreensão, para entender bem o que e como deve fazer programa, para que ele execute tarefas que solucionem o problema proposto; Conhecimentos de lógica e matemática, pois, muitas questões de programação envolvem cálculos.

Programação de Computadores: Compreender as dificuldades de aprendizagem dos alunos

Ambrósio et al (2011) explanam sobre os motivos dos alunos da disciplina de programação em cursos superiores encontrarem dificuldade em tal componente curricular. São explicadas várias razões para os insucessos na programação, além de ser demonstrado quais habilidades são usadas no processo de programação. Tais razões foram obtidas na realização de uma pesquisa de campo com professores e alunos da disciplina de introdução a programação I, da Universidade Federal de Goiás (Brasil) e da Universidade do Minho (Portugal), e são: fazer uma análise minuciosa do problema, para chegar a solução e abstração, para resolver o problema de forma mais generalizada, podendo ser usada para outras questões.

Uma primeira habilidade cognitiva reporta-se à capacidade de sair de uma análise mais holística dos problemas do cotidiano e fazer uma leitura mais analítica dos mesmos, realizando inclusive, um planejamento sequencial. (AMBRÓSIO et al, 2011)

Dessa forma, para chegar a uma solução para o problema que foi apresentado ao aluno, ele deve analisá-lo de forma minuciosa e conseqüentemente, deve planejar em passos o

que deve ser feito para obter a solução da questão. Assim, o discente esquematiza o programa a ser desenvolvido e sabe exatamente, o que precisa ser feito que seu código seja finalizado.

Uma segunda habilidade cognitiva mencionada pelos professores é a capacidade de abstração. Por seu intermédio, espera-se que os alunos passem do mundo concreto para uma outra realidade mais semântica e simbólica, formulando ou representando um problema de forma mais abstrata e menos presa aos pormenores e elementos singulares concretos. (AMBRÓSIO et al, 2011)

Os alunos não aprendem apenas a solucionar cada problema que lhes é apresentado, mas passam a identificar os tipos de problema para que, quando solucionar um, se tornar capaz de resolver outros semelhantes ao primeiro.

2.4.2 Ensino de programação nas Universidades

O ensino de programação é aplicado nas universidades, nos cursos da área da computação, tais como: Ciências da computação, Licenciatura em computação, análise de sistemas, entre outros. Essa disciplina também é ensinada em outros cursos da área de exatas, como engenharia elétrica, física e matemática.

Nos anos 70, a linguagem Pascal(1970), foi criada pelo suíço Niklaus Wirth(1934), que tinha como principais finalidades: ser utilizada para o desenvolvimento de softwares e para o ensino de linguagens de programação. A partir do Pascal outras linguagens foram desenvolvidas, por isso, quem aprende programação fazendo uso desse tipo de linguagem, terá facilidade de compreender outras. Dessa forma, elas são utilizadas como linguagem introdutória em programação.

Ao longo das décadas, várias linguagens foram desenvolvidas, dessa forma, algumas foram melhor aceitas pelo mercado e pela academia, enquanto outras não tiveram tanto êxito. Porém, aquelas que tiveram sucesso, acabaram sendo utilizadas em disciplinas de programação em cursos superiores de programação. Sendo assim, muitas linguagens que foram criadas há décadas atrás, com o passar dos anos já foram ensinadas, algumas permanecem, a exemplo de C(1973) e C++(1983), outras nem tanto como: COBOL(1959) e FORTRAN(1957).

A programação no ensino superior existe como parte integrante de várias matérias, inclusive aquelas com outro foco, mas que demande o seu uso no desenvolvimento de atividades ao longo da disciplina. Para ilustrar isso, serão apresentadas a seguir a grade curricular obrigatória dos cursos de Ciências da Computação da Universidade Federal de Campina Grande (UFCG) e da Universidade Federal do Maranhão (UFMA).

Segundo o Projeto Político Pedagógico de Ciências da Computação da UFCG (1999), as disciplinas próprias da computação presentes na grade curricular obrigatória, são:

Disciplinas de Ciências da Computação da Universidade Federal de Campina Grande
• Programação I;
• Programação II;
• Paradigmas de Linguagens de Programação Algoritmos
• Teoria dos Grafos
• Teoria da Computação
• Estrutura de Dados e Algoritmos
• Lógica Matemática Organização e Arquitetura de Computadores
• Introdução à Computação
• Organização e Arquitetura de Computadores I Tecnológicas
• Gerência da Informação
• Engenharia de Software I
• Sistemas de Informação I Estágio
• Laboratório de Programação I
• Laboratório de Programação II
• Laboratório de Estrutura de Dados e Algoritmos
• Laboratório de Organização e Arquitetura de Computadores
• Laboratório de Engenharia de Software
• Laboratório de Interconexão de Redes de Computadores
• Projeto em Computação I
• Projeto em Computação II
• Análise e Técnicas de Algoritmos
• Métodos e Software Numéricos Humanidades
• Informática e Sociedade
• Compiladores
• Redes de Computadores
• Interconexão de Redes de Computadores
• Sistemas Operacionais

• Banco de Dados I
• Banco de Dados II
• Sistemas de Informação II
• Inteligência Artificial I

Tabela1. Disciplinas de ciências da computação da Universidade Federal de Campina Grande

Fonte:

Segundo o Plano Político Pedagógico da UFMA (2007), a grade curricular possui as seguintes disciplinas relacionadas a computação, são elas:

Disciplinas relacionadas a computação do curso de Ciências da Computação da Universidade Federal do Maranhão
Algoritmos I
Arquitetura de Computadores
Compiladores
Estrutura de Dados I
Estrutura de Dados II
Introdução a Computação
Linguagem de Programação
Linguagens Formais e autômatos
Matemática Discreta e Lógica
Paradigmas de Programação
Teoria da Computação
Banco de Dados I
Engenharia de Software I
Inteligência Artificial
Processo de desenvolvimento de Software
Redes de Computadores
Sistemas Operacionais I

Tabela2. Disciplinas relacionadas a computação do curso de ciências da computação da Universidade Federal do Maranhão

Fonte:

Analisando as disciplinas apresentadas das duas instituições federais de ensino superior, é possível perceber que as disciplinas relacionadas a programação são inúmeras,

podendo diferir inclusive em quantidade, o que mostra que os conteúdos vistos podem ser diferentes no mesmo curso de universidades distintas. Isso acaba tornando difícil definir um padrão de conteúdos gerais em componentes curriculares que envolvam a programação, com exceção das disciplinas introdutórias, onde os conteúdos são praticamente os mesmos.

Para comprovar isso serão expostas as disciplinas introdutórias de programação que existem na Universidade Federal de Campina Grande (UFCG) e Universidade Federal do Maranhão (UFMA), são:

- Linguagem de Programação, segundo o Plano Político Pedagógico do Curso de Ciências da Computação da UFMA (2007), “Estudo detalhado de uma linguagem de programação. Estrutura da linguagem. Comandos e declarações. Tipos de dados. Manipulação de Arquivos. Aplicações”.
- Programação I, segundo o Plano Político Pedagógico da UFCG (1999), “Resolução de problemas. Modelagem do mundo real. Algoritmos. Noções básicas de programação. Tipos de dados. Entrada e Saída. Decisões. Laços. Vetores e Matrizes. Recursividade. Alocação dinâmica de memória. Técnicas básicas para boa programação. Tratamento de erros. Testes”.

Observa-se, assim, que os conteúdos vistos nas universidades são basicamente os mesmos. Elas têm como foco o ensino de conceitos básicos dessas disciplinas, como: Algoritmos, estudo de declarações, tipos de dados, entre outros assuntos.

Normalmente, as linguagens de programação escolhidas para ensino são aquelas que possuem um grande apelo comercial, que são utilizadas pela maioria das empresas desenvolvedoras de *softwares* ou que façam uso da programação para realizar determinada tarefa. Segundo o ranking da IEEE(2013) de linguagens de programação mais utilizadas, que considera fatores de crescimento, como: interesse demanda de empregos ou popularidade em fóruns e redes sociais; as linguagens que estão entre as quatro que mais crescem, são: Java, C, C++ e Python.

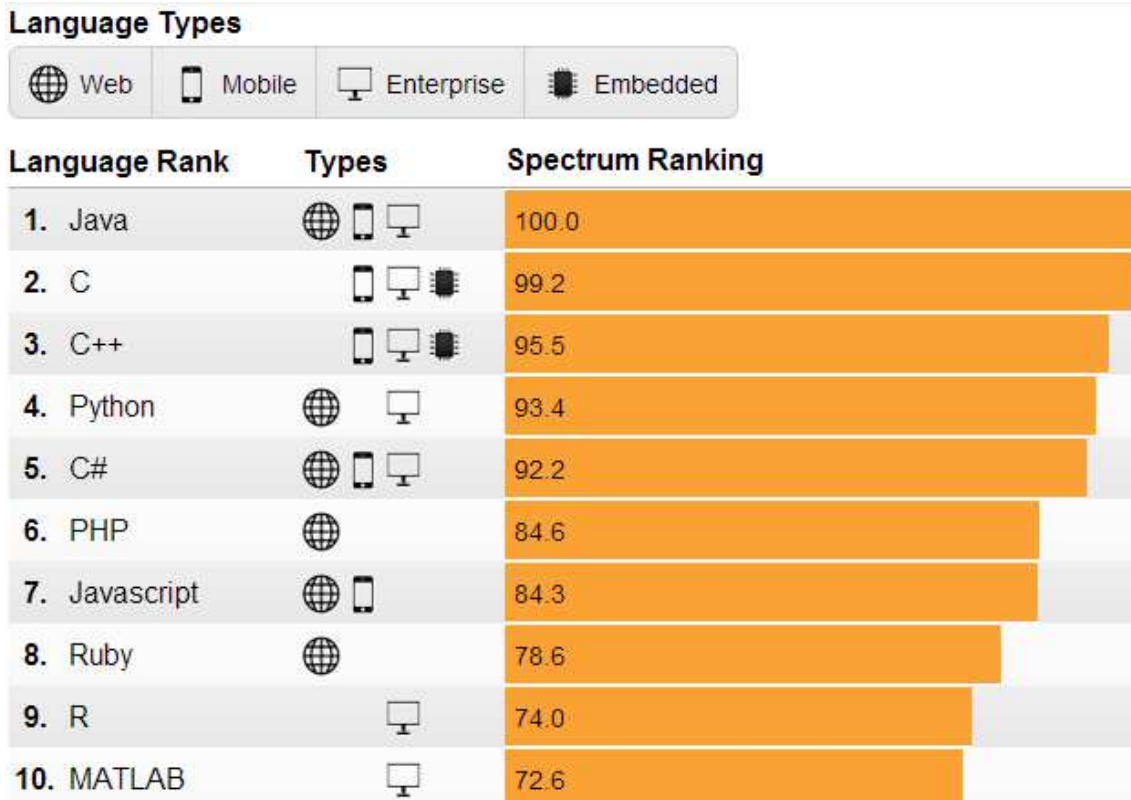


Figura 1. Ranking das linguagens que estão em crescimento.

Fonte: <http://www.equilatera.com.br/java-lidera-ranking-das-linguagens-de-programacao-mais-utilizadas-mundo>

Outro fator importante para o ensino dessas linguagens, como Java, é por ela ser orientada a objetos. Segundo Farinelli(2007), tal paradigma busca representar o mais fiel possível as situações do mundo real, sendo que o nosso cotidiano é marcado pela interação de vários objetos entre si. Assim, a orientação a objetos entende que um programa não são apenas um conjunto de processos, mas sim uma coleção de objetos que interagem uns com os outros.

Além disso, esse paradigma se caracteriza na divisão do programa, o que acaba contribuindo para uma boa organização do código. Ele também busca melhorar a reusabilidade, como também a extensibilidade do programa(Farinelli,2007). Após o código ser finalizado, ele é compilado e todas suas partes são unidas, constituindo assim um programa.

Apesar de as linguagens de programação que servem como objeto de estudo em cursos superiores de computação serem diferentes entre cursos de diferentes instituições, os conceitos básicos que são ensinados são os mesmos, como: subrotinas, funções, laços de repetição, estruturas, condicionais e variáveis. Embora as linguagens sejam distintas, os

alunos de qualquer uma dessas instituições pode utilizar os conceitos de uma já aprendida e adaptar apenas a sintaxe, se desejarem aprender linguagens adicionais.

2.4.3 Programação nas escolas

A disciplina de linguagem de programação não está presente na grade curricular das escolas públicas no Brasil. Mesmo assim, várias instituições públicas no país oferecem o ensino dessa matéria. Nas escolas em que essa matéria está presente, normalmente são utilizadas linguagens de programação educacionais, que são mais simples e de fácil compreensão. Isso permite que o primeiro contato com a disciplina seja proveitoso e motivante.

Essas linguagens contribuem para introduzir os alunos nos conceitos básicos de programação, pois eles dispõem de elementos de linguagens de programação de alto nível (como Java, C++, C e Python), tais como: variáveis, laços de repetição e estruturas condicionais. Além disso, elas podem trabalhar conceitos de outras matérias, a exemplo do: Logo(1967) e Scratch(2007).

2.4.3.1 Linguagem educativa: Logo

Logo é uma linguagem de programação criada pelo educador matemático Seymour Papert, em 1967, no MIT (Massachusetts Institute of Technology). Ela foi desenvolvida, a partir de uma filosofia de inspiração piagetiana, na qual a criança aprende explorando o ambiente que está utilizando. Segundo Valente (1998 apud FERUZZI, 2001, p.1), essa a linguagem de programação apresenta características relacionadas a computação, tais como: exploração de atividades espaciais, fácil terminologia e capacidade de criar novos procedimentos. Assim, essa linguagem aproxima os alunos de diversos conceitos de programação, como também exercita conhecimentos de matemática, a partir de seus próprios recursos. Além disso, ela é utilizada para ensinar programação, pois ela é de fácil compreensão e possui comandos de fácil entendimento.

Logo consiste em uma mascote (tartaruga) localizada inicialmente no centro da tela, que responde às entradas do usuário. Por ser do tipo interpretada e interativa, após o código ser digitado, pode-se visualizar o resultado do programa imediatamente. Assim, se o objetivo não for alcançado, o aluno será notificado na tela do programa, sobre o erro que impossibilitou a realização da ação pretendida. Em seguida, ele poderá buscar a solução para seus erros, melhorando o aprendizado.

Graças aos movimentos da tartaruga, a linguagem pode ser utilizada para o ensino de matemática, transmitindo ou colaborando no aperfeiçoamento de conhecimentos matemáticos, como ângulos e plano cartesiano. O desenho da forma é feito pela mascote, que executa seu movimento, de acordo com as entradas digitadas pelo usuário no programa. Para fazer as formas geométricas, os alunos precisam revisar sobre plano cartesiano, tendo noção dos valores de x e y na tela do programa, para assim saberem quais os valores de cada eixo o traço deve seguir. Também torna possível trabalhar com o ensino de programação, possibilitando a aprendizagem de conceitos típicos da programação, como o uso de comandos, operadores, tipos de dados, estruturas condicionais e laços de repetição.

Existem vários ambientes de programação genéricos ao Logo, como o: Kturtle. Para ser explicado melhor sobre comandos, variáveis, laços de repetições; será utilizada esse software, que foi desenvolvido pela KDE(1996), em 2003.

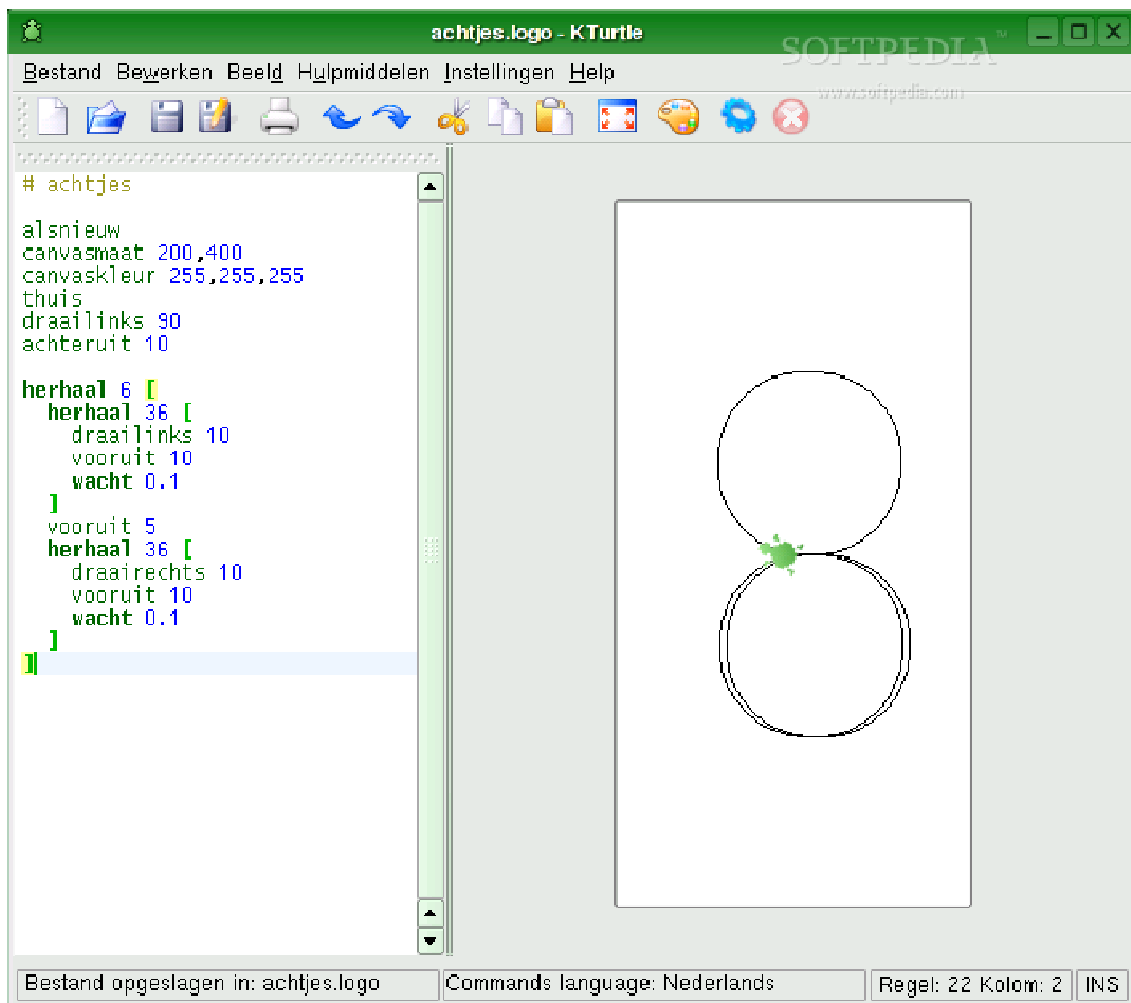


Figura2.Ambiente do Kturtle.

Fonte: <http://linux.softpedia.com/get/Education/KTurtle-2303.shtml>

No Kturtle, como em outras linguagens de programação, também é necessário declarar as variáveis que serão utilizadas, que são nomes que não fazem parte das palavras reservadas dela. Nessa linguagem, a variável deve ser declarada sempre precedida do símbolo “\$”, como ilustra o exemplo: \$nome. Se ela for referente a um valor, poderá ser inicializada com um número, como: \$valor=10.

Os laços de repetição nessa linguagem são mais simples, tendo apenas o nome “repita” seguido de chaves e dentro delas existem os comandos que devem se repetir executar determinada tarefa, como exemplifica código para fazer um quadrado:

```
Repita 4{  
  Para frente 100  
  Para esquerda 90  
}
```

É possível entender, portanto, que diversos conceitos de programação que são usadas em linguagens que não tem uma finalidade educacional, como também são fáceis de compreender aqueles comandos usados em linguagens de programação educacionais.

2.4.3.2. Linguagem educativa: Scratch

O Scratch(2007) foi criado por Mitchel Resnick, no Media Lab, em 2007, no MIT (Massachusetts Institute of Technology). O programa também possui uma mascote, que é um gato, cujo uso é opcional, que executa as ações desejadas pelo usuário.

A maioria das linguagens que não são educacionais, são baseadas em texto, com inúmeros comandos que requerem um pouco de estudo do paradigma relacionado. Cada comando tem sua funcionalidade específica e um programa consiste em uma série desses em um passo a passo. No scratch, a programação é feita a partir de blocos, os usuários devem conectar os blocos que possuem as funcionalidades que serão necessárias para a execução do programa que está sendo desenvolvido. Existem várias categorias de blocos de funcionalidades, cada uma possuindo uma cor, e cada uma delas tem diversas funções. As categorias são: aparência, controle, sensores, operadores, variáveis, som, caneta e movimento.



Figura3. Categorias dos blocos de funcionalidades do Scratch.

Fonte:<http://carlosrafaelgn.com.br/Scratch/>

Essa linguagem de fácil usabilidade pode ser usada com várias finalidades na área da educação. O seu uso mais comum é o desenvolvimento de jogos, principalmente os educativos. Os jogos que são produzidos podem ser compartilhados na página do jogo, através da conta do usuário. Assim, é possível ver como os alunos utilizam essa ferramenta e quais os tipos de jogos que são produzidos. E através disso pode acontecer uma troca de experiências e de conhecimentos, já que cada desenvolvedor observa como o outro utilizou determinados comandos para atingir seus objetivos e reusa códigos existentes para futuros projetos.

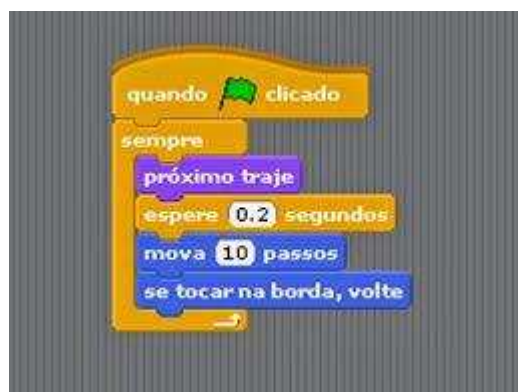


Figura4. Blocos de comandos do Scratch conectados.

Fonte: <http://www.aldneyemidio.com/2014/08/programas-para-aprender-logica-de.html>

A utilização do Scratch pode ser feita para fixação de conceitos de várias disciplinas pois, segundo o MIT (2014), a partir do Scratch os jovens aprendem a pensar e refletir de forma diferenciada. No caso da matemática, por exemplo, O Scratch ajuda os jovens a aprender a pensar de maneira criativa, refletir de maneira sistemática e trabalhar de forma colaborativa — habilidades essenciais para a vida no século 21. professores podem ensinar aos alunos como pode ser feita uma forma geométrica, a exemplo do quadrado e os discentes terão que utilizar alguns blocos que possuem funções que façam desenho de linha, estimulando conhecimentos sobre ângulos, para assim saber em qual posição a linha deve ser feita. Em alguns dos próximos capítulos serão mostradas como alguns professores a utilizam em laboratórios de informática no Brasil.

Os alunos aprendem ideias matemáticas e de computação. À medida que criam programas em Scratch, eles aprendem conceitos computacionais nucleares como iteração e condicionais (conditionals). Também desenvolvem a compreensão de conceitos matemáticos importantes como os de coordenada, variável e números aleatórios. MIT (apud SILVA e TAVARES, 2012, p.9)

A partir do desenvolvimento de projetos no Scratch, pelos alunos, eles vão aprendendo conceitos de programação de forma geral, como condições e laços de repetição. Além disso, eles reforçam conceitos matemáticos que já estudaram em sala de aula. O Scratch é, portanto, um tipo de linguagem de programação simples, que através de seus comandos pode possibilitar aos alunos desenvolverem seus programas com recursos como sons e imagens, atraindo seu interesse tanto por programar como pela disciplina a ser estudada através desse ambiente. Para o alunos, as aulas que usam essa linguagem de programação são uma oportunidade de ir além do conteúdo, já que eles podem produzir algo a partir do que foi aprendido.

2.4.4. O estudo de programação relacionado a outras aprendizagens

A literatura é praticamente unânime ao defender que a aprendizagem de programação está associada a outras disciplinas. Essa combinação é uma forma de interdisciplinaridade e tem o objetivo de colaborar para o aprendizado do discente. Segundo Germain (1992 apud FAZENDA, 2008, p.46), a interdisciplinaridade condiz a utilização de pelo menos duas disciplinas em conjunto, sendo que uma se relaciona com a outra.

O trabalho interdisciplinar da programação com outras disciplinas depende muito do conhecimento sobre a matéria e da criatividade do professor, que passa a ser responsável por

desenvolver atividades que explorem os assuntos pretendidos por ele, atendendo as necessidades de aprendizagem dos alunos, como será mostrado a seguir o ensino de desenvolvimento de jogos de matemática com o Scratch.

Oliveira et al (2013) colocou em prática o ensino de programação baseado na linguagem Scratch, com uma turma do 9º ano de uma escola de Garanhuns-PE. Após os alunos aprenderem conceitos acerca de programação, necessários para desenvolver os projetos que os professores passassem, eles colocaram tal conhecimento em prática na criação de jogos, abordando assuntos de matemática, como operações aritméticas e geometria.

De acordo com a avaliação feita por Oliveira et al (2013), foi verificado que os discentes demonstraram entusiasmo em desenvolver jogos e conseguiram compreender bem o uso da ferramenta. Além disso, os alunos externaram que passaram a gostar de matemática depois de utilizá-la nas aulas de programação.

O estudo de programação aliado a uma disciplina escolar não é para simplesmente, aprender a programar, mas também, como foi mostrado nos resultados da prática de Oliveira (2013) e outros autores, para motivar e despertar o interesse, principalmente, do componente curricular estudado em sala de aula. Então, a partir do estudo de programação, os discentes podem aprender e reforçar tais matérias.

2.4.4.1. O estudo de programação e o desenvolvimento de habilidades úteis no processo de aprendizado

No estudo de programação, além de os discentes aprenderem conceitos próprios da disciplina e passarem a dominar uma determinada linguagem, eles podem utilizar tais conhecimentos para obter êxito em outras disciplinas. Essa matéria serve também para exercitar e aprimorar habilidades que podem ser úteis no dia-a-dia dos alunos, tanto dentro, como fora da escola.

Aprender a programar é algo complexo, e requer esforço, perseverança e uma abordagem espacial no que concerne à forma de estudo e de ensino. Há um conjunto de habilidades envolvidas que vão muito além de, saber a sintaxe da linguagem”
(GOMES et al, 2008)

O ato de programar é algo realizado de forma minuciosa, e seu estudo difere de disciplinas escolares, tipicamente baseadas na leitura e na memorização. Programar vai além de simplesmente entender a linguagem que está usando para criar o programa, escrever o código na IDE(Integrated Development Environment) e compilá-lo. Ela envolve uma série de

passos a serem seguidos, para assim o aluno saber o que o seu código deve fazer, como realizá-lo e constatar se o seu resultado é aquilo que ele queria alcançar. Dessa forma, o aluno acaba exercitando muito a mente, fazendo com que os alunos obtenham novos significados daquilo que sabem, os modificando quando for apropriado.

De forma geral, programar é visto como uma forma de resolução de problemas que pode ser decomposta em quatro fases: entender o problema, implementar um plano e verificar se está certo ou se é adequado o resultado final atingido (AMBRÓSIO et al, 2011)

Para se resolver um problema através de programação, é necessário entendê-lo para posteriormente saber como deve proceder, refletindo sobre o que o código deve fazer, quais comandos deverão ser utilizados e como será a estrutura do aplicativo.

Ao tentar encontrar uma solução, para um dado problema, o aluno não fica ligado a dados específicos; pelo contrário, busca abstrair da solução, o “processo”, possível de ser futuramente aplicado em várias instâncias (AMBRÓSIO et al, 2011).

No momento quando o problema é passado para os alunos, eles devem ser motivados a encontrar soluções menos específicas, mais generalizadas, para que cada solução possa servir para outras situações semelhantes. A ideia é que, quando novas questões forem passadas para os discentes, eles verifiquem se seria possível solucioná-la de forma similar a outras que ele já resolveu anteriormente. Isso só seria possível se o aluno entender o que o código deve fazer, tendo raciocinado para chegar às soluções, o que requer um processo de concentração significativo.

As linguagens de programação têm uma sintaxe muito complexa. A maioria das linguagens de programação foi desenvolvida com um propósito e não para suportar a sua aprendizagem, sendo extensivas e com muitos detalhes sintáticos complexos e características específicas a memorizar. Essa complexidade requer que os estudantes tenham de se concentrar simultaneamente na tarefa de construção do algoritmo e na tarefa de recordar regras sintáticas (GOMES et al, 2008).

O ato de se concentrar deve acontecer pois serve para definir como o programa deve ser feito, para escrever o código de forma correta, para colocar os comandos certos, na ordem adequada, e definir métodos e funções. Enfim, é necessário se concentrar para minimizar a quantidade de erros de cada código, abstraindo soluções através de modelos mentais ou formais, antes de apresentá-las concretamente, no aplicativo em funcionamento.

A programação envolve um elevado nível de abstração. A aprendizagem de programação requer uma hierarquia de competências de que se destacam um elevado nível de abstração, generalização, transferência, e pensamento crítico (GOMES, 2008).

Enfim, para programar não é necessário apenas utilizar conhecimentos provenientes de disciplinas de programação, mas também utilizar habilidades como as citadas anteriormente. O uso das mesmas na programação possibilita o seu aprimoramento, e habilita, no caso do ensino dessa disciplina, os discentes para usá-las também no seu cotidiano.

3. O ensino de programação e relações com a educação

O ensino de programação, como já foi visto anteriormente, não apenas fornece conhecimentos sobre essa disciplina ou acerca da computação, mas é ela que pode trazer várias contribuições, caso ela seja colocada em prática nas escolas, tanto para os alunos como para a própria escola. Assim, pode colaborar para efetivação do uso laboratórios, como também pode promover uma aprendizagem significativa.

3.1. O ensino de programação para efetivação do uso dos laboratórios

A implantação dos laboratórios nas escolas públicas ganhou força a partir do PROINFO em 1997, que tinha o objetivo de levar o uso do computador para as escolas, para que eles fossem utilizados de forma educativa. Porém, a ausência de um planejamento quanto ao ensino de disciplinas de computação, tornando-as obrigatórias, fez com que a disponibilização de recursos do PROINFO fosse inefetiva, resultando em apenas 24% dos laboratórios sendo utilizados, como mostra a pesquisa TIC e Educação (2011). Além disso, nas escolas que fazem uso deles, é muito comum ausência de professores capacitados para o ensino de conteúdos relacionados a computação.

Por causa disso, os alunos acabam não enxergando a utilização dos laboratórios como aulas de fato. Segundo Poletto (2009, apud CRUZ; SILVA, 2013), “eles acham que ir para o laboratório é um lazer e não uma atividade da escola”. Muitos professores também sentem dificuldade para usar o computador, o que acaba criando uma barreira para os alunos.

Como o professor é um imigrante da era digital, o professor não tem aquela rapidez, aquela sagacidade, aquele interesse que o aluno tem. Poletto (POLETTTO, 2009, apud CRUZ e SILVA, 2013).

Dessa forma, muitas escolas não procuram mostrar a importância que a utilização dos laboratórios de informática pode ter para o aluno, para prepará-lo para viver num mundo onde

as tecnologias estão cada vez mais presentes e o conhecimento em utilizá-las é cada vez mais necessário.

Para a efetivação da utilização dos laboratórios, é preciso que seja estabelecida uma certa regularidade no uso desses ambientes, o que poderia ser feito com a instauração do ensino de uma disciplina que trabalhe tanto o lado da computação como a parte educativa, como a programação. As escolas que a põem em prática, trabalham conceitos de disciplinas vistas em sala de aula, além explorar o aprendizado dessa matéria da computação.

Isso é comprovado na experiência de Oliveira et al (2013), que realizou o ensino de programação com a ferramenta Scratch, dando ênfase ao desenvolvimento de jogos de matemática. Conforme vimos anteriormente, os autores aliaram o ensino de uma disciplina própria da computação com uma matéria que é ensinada em sala de aula. Há ainda a barreira da qualificação: embora existam profissionais graduados na área de computação, como os licenciados em computação, habilitados para trabalhar o ensino de matérias como programação nas escolas, muitos estados não oferecem vagas em concursos públicos para esses.

No concurso do estado da Paraíba, em 2012, de acordo com o edital (<http://www2.ibfc.org.br/concurso/seepb-1231/docs/seepb-01-2012-edital.pdf>), para o cargo de professor do ensino básico 3, com 2.000 vagas entre 10 disciplinas, não existia uma vaga para professor de computação, conforme a Tabela.

CARGO	DISCIPLINAS	TOTAL DE VAGAS	PRÉ-REQUISITOS
PROFESSOR DA EDUCAÇÃO BÁSICA 3	Língua Portuguesa	348	Licenciatura Plena em Letras – Habilitação em Língua Portuguesa
	Língua Inglesa	170	Licenciatura Plena em Letras – Habilitação em Língua Inglesa
	Arte	20	Licenciatura Plena em Arte ou Educação Artística
	História	216	Licenciatura Plena em História
	Geografia	244	Licenciatura Plena em Geografia
	Biologia	186	Licenciatura Plena em Ciências Biológicas *
	Química	178	Licenciatura Plena em Química*
	Física	178	Licenciatura Plena em Física*
	Matemática	400	Licenciatura Plena em Matemática*
	Educação Física	60	Licenciatura Plena em Educação Física

Tabela3. Cargos e graduações exigidas para concurso para professor de 2012.

Fonte:<http://www2.ibfc.org.br/concurso/seepb-1231/docs/seepb-01-2012-edital.pdf>

Os licenciados em computação são os profissionais que podem exercer o ensino de programação nas escolas, pois estudam na graduação tanto disciplinas próprias da computação, que os habilita para o ensino de conteúdos na sua área, como pedagógicas, que dão a base teórica para que tais assuntos sejam ensinados. De acordo com o Plano Político Pedagógico do curso de licenciatura em computação da UFPB (2006), a graduação referida tem como objetivos formar professores para o ensino de computação e informática para os ensinos fundamental, médio, profissional e em empresas também. Eles podem, dessa forma, atender as necessidades das escolas quanto ao ensino de computação e informática.

O curso de Licenciatura em Computação tem por objetivo formar educadores para o ensino de computação e informática das escolas das redes pública e particular e do ensino fundamental e médio; da educação profissional, para a qualificação para o trabalho; das empresas, onde a computação constitui-se na base da formação para treinamento e educação corporativa. O curso visa atender à demanda crescente das escolas e organizações que estão em processo de modernização tecnológica. (PLANO POLÍTICO PEDAGÓGICO DE LICENCIATURA EM COMPUTAÇÃO DA UFPB, 2006)

O documento ainda defende que os licenciados, devem ser professores que ensinam conteúdos da ciência da computação como uma disciplina essencial para os alunos.

O Curso de Licenciatura em Computação visa formar professores na área de computação, capazes de trabalhar os conteúdos da ciência da computação, necessários e significativos para os anos finais do ensino fundamental e médio. (PLANO POLÍTICO PEDAGÓGICO DE LICENCIATURA EM COMPUTAÇÃO DA UFPB, 2006)

De acordo o Plano Político Pedagógico do curso de Licenciatura em Ciências da Computação da UFPB (2006), as disciplinas obrigatórias básicas do mesmo são:

1. Conteúdos Básicos Profissionais			
1.1 Conteúdos Básicos da Ciência da Computação			
Disciplinas	Créditos	Carga Horária	Pré-requisitos
Introdução ao Computador	04	60	-
Introdução à Programação	04	60	-
Linguagens Declarativas	04	60	Ling de Programação
Linguagem de Programação	04	60	Introdução à Programação
Programação Orientada a Objetos	04	60	Linguagem de Programação
Arquitetura de Computadores I	04	60	Introdução à Programação
Arquitetura de Computadores II	04	60	Arquitetura de Computadores I
Estrutura de Dados	04	60	Linguagem de Programação
Teoria da Computação	04	60	Lógica Aplicada à Computação
Análise e Projetos de Sistemas	04	60	Banco de Dados
Total	40	600	
1.2 Conteúdos Básicos da Matemática			
Matemática Elementar	04	60	-
Lógica Aplicada à Computação	04	60	Matemática Elementar
Cálculo I	06	90	-
Cálculo II	06	90	Calculo I
Probabilidade e Estatística	04	60	Cálculo II
Cálculo Vetorial e Geometria Analítica	04	60	-
Álgebra Linear	04	60	Cálculo Vetorial e Geometria Analítica
Total	32	480	
1.3 Conteúdos Básicos Tecnológicos			
Sistemas Operacionais	04	60	Arquitetura de Computadores II
Engenharia de Software	04	60	Análise e Projetos de Sistemas

Redes de Computadores	04	60	Sistemas Operacionais
Inteligência Artificial	04	60	Linguagem de Programação
Banco de Dados	04	60	Modelagem de Dados
Modelagem de Dados	04	60	Estrutura de Dados Programação Orientada a Objetos
Sistemas Multimídia	04	60	-
Total	28	420	
1.4 Conteúdos Básicos de Formação Humanística			
Ética Profissional	03	45	
Computadores e Sociedade	02	30	-
Total	05	75	
1.4 Conteúdos Básicos de Formação Pedagógica			
Fundamentação Antropo-Filosófica da Educação	04	60	
Fundamentação Sócio-Histórica da Educação	04	60	
Fundamentação Psicológica da Educação	04	60	
Política e Gestão da Educação	04	60	
Didática	04	60	Fund Antropo-Filos da Educação Fund Sócio-Hist da Educação Fund Psic da Educação
Total	20	300	
Estágio Supervisionado I	03	45	Didática
Estágio Supervisionado II	08	120	Estágio Supervisionado I
Estágio Supervisionado III	08	120	Estágio Supervisionado II
Estágio Supervisionado IV	08	120	Estágio Supervisionado III
Total	27	405	

Tabela4. Disciplinas obrigatórias do curso de Licenciatura em ciências da Computação UFPB.
Fonte: http://www.ufpb.br/sods/consepe/resolu/2006/Rsep69_2006.htm

Os objetivos desse curso, citados anteriormente, defendem que os graduados devem colocar em prática nas escolas conteúdos da ciência da computação, onde a instrução de programação faz parte. Além disso, a grade curricular obrigatória dessa graduação aponta que os discentes estudam disciplinas tanto relacionadas a programação, como a educação. Os conhecimentos dessas duas áreas combinados podem servir para o ensino de programação, pois os licenciados podem ensinar essa matéria a partir do que aprenderam nas matérias de programação, e usar conhecimentos teóricos pedagógicos para dar suporte a essa instrução.

Por não existir efetividade do ensino nos laboratórios, a utilização desses ambientes depende muito de cada escola. Entende-se que é necessário que o ensino de uma disciplina relacionada à computação, como programação, seja obrigatório para as escolas como obrigatória. Para que essa implantação seja possível, os estados precisam dar oportunidades, abrindo vagas em concursos públicos, de maneira que os profissionais qualificados possam exercer suas profissões.

3.2. A programação e a aprendizagem significativa

Como já visto, ensino de programação engloba algumas disciplinas. Quando essa matéria é aplicada nas escolas, tipicamente tem como estratégia usar outras disciplinas para o seu ensino, justamente porque sua instrução acontece de forma significativa.

A associação de conteúdos novos, vistos atualmente e os já existentes na estrutura cognitiva é o que configura uma aprendizagem significativa. Ela é realizada, dessa forma, através da ligação entre duas ou mais disciplinas, como: a programação e outras matérias. A interdisciplinaridade acontece principalmente no momento da resolução de exemplos e exercícios, atividades típicas de programação, especialmente se envolverem outros assuntos.

Por exemplo, para resolver uma questão que peça que os estudantes façam um programa que mostre o quadrado de um valor, é necessário utilizar de conceitos matemáticos (exponenciação), além de conhecimentos de programação (que comandos utilizar, quais variáveis declarar, como deve ser pedido o valor ao usuário para saber seu quadrado, como deve ser feito o cálculo para saber o quadrado de um número e como fazer para apresentar o resultado na tela). Nesse caso, matemática e programação se interligariam para que o aluno chegasse ao resultado final: um programa que apresenta o quadrado de um número digitado.

Quando um assunto é visto em sala de aula e logo depois na aula de programação, as informações obtidas anteriormente e as novas se relacionam, “ancorando”-se na estrutura cognitiva do indivíduo, e o que foi aprendido anteriormente pode ser considerado por subsunção. Porém, o conteúdo de programação também precisa ser previamente apresentado ao aluno de forma mais simplificada, por meio de organizadores prévios, que servem para que eles se familiarizem com os conteúdos antes da aplicação das questões interdisciplinares.

A aprendizagem significativa processa-se quando o novo material, ideias e informações, que apresentam uma estrutura lógica, interagem com conceitos relevantes, claros e disponíveis na estrutura cognitiva, sendo por eles assimilados, contribuindo para sua

diferenciação, elaboração e estabilidade (MOREIRA; MASINI, 1982). Dessa forma, os conteúdos das duas disciplinas ensinadas nas aulas de programação - sejam elas a própria programação e a matéria escolar - precisam ser apresentados de forma simples e superficial anteriormente para os discentes, para que antes do aprendizado delas no laboratório de informática seja simplificado, ajudando na compreensão do que será estudado adiante.

Assim, o ensino de programação aliado a alguma disciplina, colocado em prática através da aprendizagem significativa, pode contribuir para o aprendizado do aluno, por buscar enraizar o conteúdo na estrutura cognitiva do aluno, a partir da interligação de conhecimentos novos e antigos, que têm relação com o conteúdo. Dessa forma, esse método visa não só uma simples memorização dos conteúdos, para que os alunos obtenham êxito na escola, mas busca assimilar o que foi transmitido em sala de aula no nível de estrutura cognitiva ao longo dos anos.

4. Considerações Finais

Ao longo desse trabalho foram apresentadas, a utilização do computador nas escolas, tanto com finalidade exclusivamente educacional, como também o objetivo de ensinar programação aliada a alguma disciplina escolar. Para tanto, foram apresentadas ideias para o ensino que podem facilitar a compreensão do conceito de programação, como também de outras disciplinas que forem usadas durante o processo de ensino, através da teoria da aprendizagem significativa.

Além disso, explicou-se que existem habilidades que os discentes usam durante o ato de programar que são úteis para cotidiano, incluindo fora da sala de aula, advogando o ensino de programação para a efetivação do uso dos laboratórios de computação. Percebe-se que, se o ensino de programação de fato acontecesse nas escolas, ele não contribuiria apenas para o aprendizado dessa disciplina pelo aluno, mas também de outras formas tanto para o discente, como para a instituição. Para o aluno, o aprendizado não fica apenas limitado à própria programação, mas se estende ao reforço de disciplinas que já fazem parte do currículo escolar. A utilização do ensino dessa matéria da computação, seria uma forma de ganhar a atenção e o interesse dos alunos, inclusive para outras disciplinas, já que a maioria dos jovens em idade escolar utilizam computadores.

Esse foi o caso da prática do ensino de programação em uma escola que incentivou os alunos a elaborar animações e jogos que usavam de conceitos matemáticos, aumentando o interesse pela matemática além de ensinar a ferramenta. As linguagens de programação educativas, como Logo e Scratch, que têm o objetivo de contribuir para o aprendizado de outras disciplinas, acabam melhorando o ensino em várias perspectivas.

Dessa forma, se o ensino de programação fosse efetivado nas escolas públicas, as linguagens de programação educativas poderiam ser utilizadas para que os alunos possam aprender como a tecnologia pode ser útil para eles, com a produção de aplicativos que tenham funcionalidades que podem ser proveitosas para os alunos. Essa utilidade se confirma pelo papel da programação como meio; nesse caso, consolidando conceitos de outras matérias, o que pode resultar em uma melhora de desempenho em qualquer disciplina trabalhada.

Nesse cenário, a aprendizagem significativa surge como uma abordagem fundamental, por buscar que o aluno entenda o que foi passado de forma minuciosa, assimile o que foi estudado e faça as relações necessárias entre conteúdos que estão sendo ou já foram estudados. É necessário que o professor procure sempre auxiliar os educandos para que eles

façam essas ligações, pois se o aluno já tiver conhecimentos relacionados ao assunto em sua estrutura cognitiva, o aprendizado acontecerá de forma mais fácil. Caso o assunto a ser estudado seja completamente novo para os alunos, o educador precisa apresentar os chamados organizadores prévios, expondo de forma simples o que será aprofundado mais adiante.

Este trabalho mostrou também a análise de uma prática de ensino de programação aliado a conceitos da língua portuguesa, o que ajudou na demonstração de elementos que fazem parte da aprendizagem significativa. Foi mostrado que é possível utilizar esse tipo de método de ensino, apesar de a experiência exposta não configurar aprendizagem significativa.

Nessa metodologia, é necessário que o aluno tenha conhecimentos do que será aprendido e o professor deve auxiliá-los nessa ligação entre assuntos para assimilar o que será estudado. Além disso, os professores devem verificar se os alunos realmente compreenderam o assunto, realizando testes que estejam em um contexto diferente daquele que já foi trabalhado antes. Portanto, o assunto é trabalhado de forma minuciosa, buscando que o aluno de fato assimile o que for estudado, retendo o que foi aprendido em sua estrutura cognitiva.

O ensino de programação, conforme exposto, difere muito de outras matérias que já fazem parte da grade curricular das escolas públicas, por se tratar de uma disciplina de muita prática e assimilação. Além disso, para programar, é necessário empregar várias habilidades, como concentração e perseverança. O discente acaba usando capacidades que são desenvolvidas ou aprimoradas, para assim, obter êxito nas matérias escolares.

Em relação à contribuição da implantação dessa disciplina na escola, muitas escolas não fazem uso do laboratório de informática, já que o ensino de matérias da área de computação não existe na grade curricular das escolas públicas. Assim, quando ocorre a utilização de tais laboratórios, ela é feita de forma educativa. Além disso, o uso dos mesmos, por ser não obrigatório, depende muito de cada professor e da importância que o mesmo deposita no uso de computadores para fins educacionais. Com o ensino da programação, tais ambientes seriam constantemente utilizados, fazendo com que as escolas sejam vistas como instituições que enxergam uma importância da aproximação dos alunos com tecnologia.

Isso só será possível quando o problema de falta de abertura de vagas em concursos públicos para profissionais licenciados em computação for resolvido. Esses são os profissionais adequados para desempenhar tal papel, já que possuem conhecimentos tanto da área pedagógica como da área da computação, ambos necessários para que colocar em prática o ensino de programação. Isso foi demonstrado na exposição do plano político pedagógico do

curso de licenciatura em computação da UFPB, que habilita estudantes com disciplinas pertencentes à pedagogia e à computação, o que contribui para que eles aprendam métodos para transmissão dos conteúdos de forma eficaz e também para o aprendizado da computação em si, em matérias como a programação.

Dessa forma, para que tal disciplina de fato seja ensinada nas escolas, é preciso não apenas exergar sua importância, mas também uma mudança de mentalidade, com a contratação de profissionais licenciados em computação e a abertura de vagas para os mesmos em concursos públicos. É necessário que os governos estaduais percebam como tal matéria pode ser importante em vários aspectos, tanto para os discentes, como para a escola.

É imprescindível, além de mostrar as contribuições do ensino de programação em escolas que o adotaram, provocar a abertura de discussões em ambientes acadêmicos sobre sua implantação e importância. Dessa forma, tal assunto deve ser sempre discutido, até que se perceba que tal ensino acontece de forma efetiva.

Para que o ensino de programação possa de fato ocorrer nas escolas, é necessário que sejam feitas pesquisas futuras mais específicas, com um foco mais pedagógico, buscando entender parâmetros curriculares, como quais séries podem incluir essa matéria, quais linguagens de programação devem ser adotadas e quais assuntos interdisciplinares podem ser abordados. Só assim os objetivos com a instrução de tal matéria serão de fato atingidos.

Referências

_____.Tic Educação 2013-Alunos: Perfil do usuário de computador e internet.Disponível em: <http://www.cetic.br/educacao/2013/alunos/B5.html>. Acessado em: 02 de fevereiro de 2014.

_____.TIC 2011: Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas.Disponível em: <http://op.ceptro.br/cgi-bin/cetic/tic-educacao-2011.pdf> . Acessado em: 29 de janeiro de 2014.

ROCHA PEREIRA JUNIOR, José Carlos; RAPKIEWICZ, Clevis Elena.O processo de ensino-aprendizagem de fundamentos de programação: Uma visão crítica da pesquisa no Brasil.Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/weirjes/2004/003.pdf> . Acessado em: 10 de fevereiro de 2015.

MERCADO,Leopoldo.Novas Tecnologia na Educação.Maceió:EDUFAL,2002.

PRENSKY,Marc.The Role of Technology in teaching and the classroom.Disponível em: http://www.marcprensky.com/writing/Prensky-The_Role_of_Technology-ET-11-12-08.pdf Acessado em: 23 de março.

Moreira M. A.,Mazini F.E.S.Aprendizagem Significativa: A teoria de David Ausubel.São Paulo:Moraes LTDA;1982.

SOARES, Maria Luísa Couto.A dimensão intencional.Disponível em: <http://books.google.com.br/books?id=8X3ltsZkuoUC&pg=PA50&dq=cart%C3%B5es+perfurados&hl=pt-BR&sa=X&ei=73pAVNfaOvfesATHyoDYDw&ved=0CD0Q6AEwAw#v=onepage&q=cart%C3%B5es%20perfurados&f=false> . Acesso em: 10 de Setembro de 2014.

SEBESTA, Robert W.Conceitos de Linguagem de Programação.Disponível.9.ed.São Paulo: Bookman, 2010.

FONSECA FILHO, Clézio.Computação: O Caminho do Pensamento e da tecnologia. Porto Alegre: EdiPUCRS, 2007.

ROBERTO LIMA,Márcio.Construcionismo de Papert e ensino aprendizagem de programação no ensino superior.Disponível em: <http://www.ufsj.edu.br/portal-repositorio/File/mestradoeducacao/Dissertacao1.pdf> . Acesso em: 16 de Setembro de 2014.

_____.Java lidera ranking das linguagens de programação mais utilizadas no mundo.Disponível em:
<http://www.equilatera.com.br/java-lidera-ranking-das-linguagens-de-programacao-mais-utilizadas-mundo> .
Acesso em: 18 de Setembro de 2014.

MARJI,Majed.Programar com Scratch: Uma introdução visual a programação com jogos,arte,ciência e matemática. São Paulo: Novatec, 2014.

GEBRAN, Maurício Pessoa.Tecnologias educacionais.Disponível em:
http://books.google.com.br/books?id=kDTpWvqZX_8C&pg=PA41&dq=software+educacional&hl=pt-BR&sa=X&ei=IUNRVMyaHIKmNseYg8AP&ved=0CDMQ6AEwAQ#v=onepage&q=software%20educacional&f=false .Acesso em: 20 de Setembro de 2014.

MOREIRA, Antonio Marco.Currículo Lattes.Disponível em:
<http://buscatextual.cnpq.br/buscatextual/visualizacv.do?metodo=apresentar&id=K4787437U0> .Acesso em: 21 de Setembro de 2014.

MASINI, Elcie Aparecida Fortes Salzano.Currículo Lattes. Disponível em:
<http://buscatextual.cnpq.br/buscatextual/visualizacv.do?metodo=apresentar&id=K4795668H1> .Acesso em: 21 de Setembro de 2014.

Da Silva, R. O.S; Márcia Andréia O. intervenção interdisciplinar através do scratch: Literatura de cordel e a programação de computadores.Disponível em:
http://3A%2F%2Fcntpp.educacao.ws%2Findex.php%2Fnovotempo%2Farticle%2Fdownload%2F8%2F11&ei=3_dvVPPuMMOdNuWlI5gE&usq=AFQjCNHcea47XMTlhZNTvetgYWC1yAenIQ&sig2=1pNvFb9N4qg4wgM-DscFng&bvm=bv.79189006,d.eXY .Acesso em: 22 de Setembro de 2014.

FAZENDA, Ivani C. Arantes.Interdisciplinaridade: História, Teoria e Pesquisa.
<http://books.google.com.br/books?id=IESxUjsjE9YC&printsec=frontcover&dq=a+interdisciplinaridade+fazenda&hl=pt-BR&sa=X&ei=xxhoVLryEomegwTMpIK4Bg&ved=0CCcQ6AEwAg#v=onepage&q=a%20interdisciplinaridade%20fazenda&f=false> . Acesso em:23 de Setembro de 2014.
COCHIK, José Leon. O computador no ensino e a limitação da consciência. São Paulo: Casa do psicólogo, 1998.

AMBROSIO L., Ana Paula et al.Programação de computadores : Compreender as dificuldades de aprendizagem dos alunos. Disponível em:
http://repositorium.sdum.uminho.pt/bitstream/1822/15554/1/Programa%C3%A7%C3%A3o%20de%20computadores_compreender%20as%20dificuldades%20de%20aprendizagem%20dos%20alunos.pdf . Acessado em: 23 de Setembro de 2014.

_____.Plano político pedagógico de licenciatura em computação.Disponível em:
http://www.ufpb.br/sods/consepe/resolu/2006/Rsep69_2006.htm . Acesso em: 25 de Setembro.

_____. Edital do concurso público para docentes.Disponível em:
<http://www2.ibfc.org.br/concurso/seepb-1231/> . Acesso em: 25 de Setembro de 2014.

PEREIRA, Leonardo. Escolas defendem o ensino de programação em escolas públicas.Disponível em: <http://olhardigital.uol.com.br/noticia/escolas-defendem-ensino-de-programacao-a-criancas-e-adolescentes/35075> . Acesso em: 26 de Setembro de 2014.

PERSKE, Rodolfo Cesar Forgiarini.Sistemas agroflorestais em pequenas propriedades em Hulha Negra.Disponível em:
www.emater.tche.br/%2Fsite%2Farquivos_pdf%2Fteses%2FMono_Rodolfo_Perske.pdf&ei=6HFzVNCYMYujNse7gLgD&usg=AFQjCNF_3QfwZrrxOvADzAeJ-u8fbO3gwQ&sig2=IIR0mO2XK2mhpiDHC8918Q&bvm=bv.80185997,d.eXY Acesso em:26 de Setembro de 2014.

SILVA, Aline Marcelino dos Santos; MORAES, Deiz Amara Silva de Souza; BATISTA, Silvia Freitas.Educação Ambiental: Scratch como ferramenta pedagógica no ensino de saneamento básico.Disponível em: www.cinted.ufrgs.br/%2Fciclo23%2Farti- aprov%2F127862.pdf&ei=9H1zVN2YIoOcgwSmjYKIDw&usg=AFQjCNEO0aQPfVscfguJp aNAIZCjLO3ZAg&sig2=GWyKA8b01xfx-ajA-B-WNg&bvm=bv.80185997,d.eXY . Acesso em: 26 de Setembro de 2014.

_____.Referência de Programação do Turtle Script.https://docs.kde.org/stable/pt_BR/kdeedu/kturtle/assignment-of-variables.html. Acesso em: 27de Setembro de 2014.

_____.Usos do laboratório de informática da escola estadual professora Calpúrnica Caldas de Amorim.Disponível em: www.sistemas.ufrn.br/shared/verArquivo?idArquivo=1522805&key . Acesso em: 30 de Setembro de 2014.

_____.Reinventando o ensino médio- Governo de Minas. Disponível em:
https://www.educacao.mg.gov.br/images/stories/publicacoes/reinventado_ensino_Medio_WEB.pdf . Acesso em: 02 de Outubro de 2014.

_____.Sobre o Scratch.<https://scratch.mit.edu/about/> .Acesso em: 03 de Outubro de 2014.

FARINELLI, Fernanda. Conceitos Básicos de programação orientada a objetos. http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1662272077_POO.pdf. Acessado em: 08 de Junho de 2015.